**Ime Prezime:**

1. Za idući kôd odrediti T(N) i O(N) (15 bodova):

int f(int \*niz, int n) {

int i = n, sum = 0;

while(i > 0) {

sum += niz[i/2];

i--;

}

while(i < n) {

sum += niz[i/2];

i++;

}

return sum;

}

1. Procijeniti klasu složenosti O() algoritma koji računa razliku dva skupa (dva niza dužine N i M sa svim različitim brojevima). Razlika skupova je niz brojeva koji sadrži sve brojeve iz prvog niza koji se ne nalazi u drugom nizu. Algoritam se može napisati na tri načina:
2. Nijedan niz nije sortiran. Svaki broj u prvom nizu uspoređujemo sa svim brojevima drugog niza.
3. Drugi niz je sortiran. Svaki broj u prvom nizu tražimo među brojevima drugog niza upotrebom binarne pretrage.
4. Oba niza su sortirana. Postavimo indeks j na početak drugog niza. Za svaki broj u prvom nizu provjeravamo redom sve brojeve u drugom nizu koji su manji ili jednaki od traženog broja uvećavajući indeks j. Indeks j se nikada ne vraća na početak drugog niza.

(15 bodova)

1. Napisati funkciju koja prima jednostruko vezanu listu (svaki element sadrži broj). Funkcija provjerava da li je lista sadrži višestruke brojeve. Funkcija vraća 1 ako su svi elementi jedinstveni ili 0 ako nisu. Kolika je složenost funkcije O()? (25 bodova)
2. Napisati funkciju koja dvije liste. Funkcija na kraj prve liste dodaje kopije svih elemenata iz druge liste. Funkcija vraća izmijenjenu listu. Kolika je složenost funkcije O()? (35 bodova)
3. Prikazati quicksort algoritam na nizu { 5, 6, 7, 3, 8, 2, 3 }. (10 bodova)